INFORMATIQUE
Sciences
Université Paris Cité

Arbres couvrants de poids minimum

M2

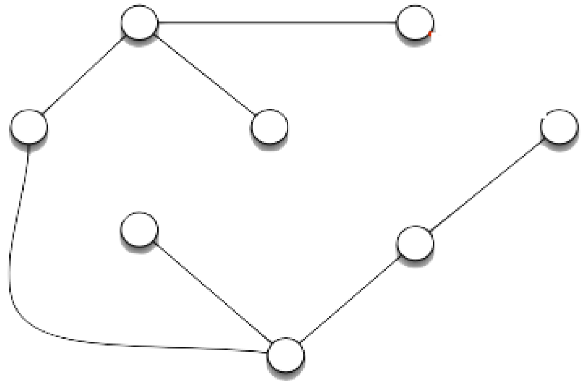
Lélia Blin

lelia.blin@irif.fr

Rappels

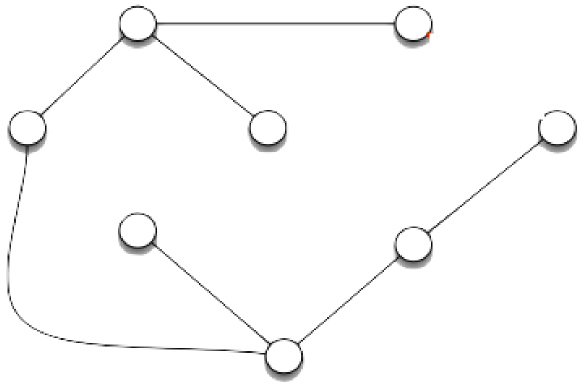
Définitions d'un arbre (connexe)

- Pour un arbre T a n sommets il y a équivalence entre les définitions suivantes :
  - T est un arbre
  - T est un graphe connexe à $n-1$ arêtes
  - T est un graphe connexe et la suppression de toute arête le déconnecte



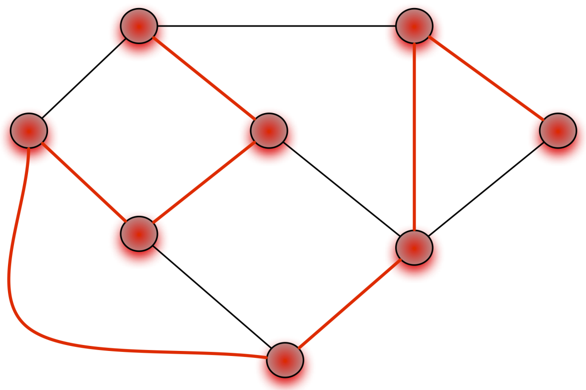
Définitions d'un arbre (cycle)

- Pour un arbre T a n sommets il y a équivalence entre les définitions suivantes :
  - T est un arbre
  - T est un graphe acyclique à $n-1$ arêtes
  - T est un graphe acyclique et l'ajout de toute arête le rend cyclique.



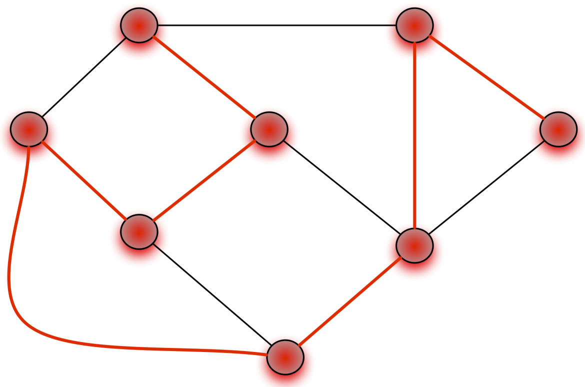
Graphe partiel

- Un graphe partiel $G'(V, E')$ d'un graphe $G(V, E)$ est:
  - Un graphe qui a les mêmes sommets que G.
  - Un graphe dont l'ensemble des arêtes E' est inclus dans E.



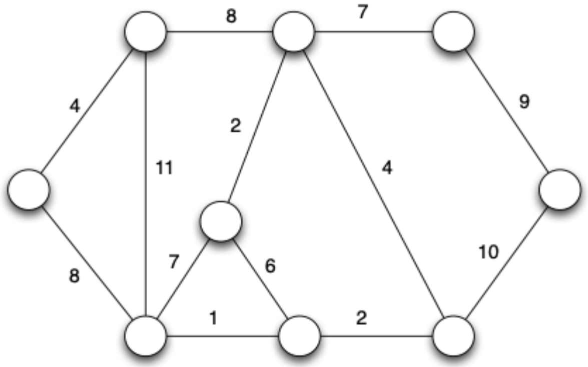
Arbres couvrants

- Un arbre couvrant T d'un graphe G(V,E) est:
  - Un graphe partiel, sans cycle.



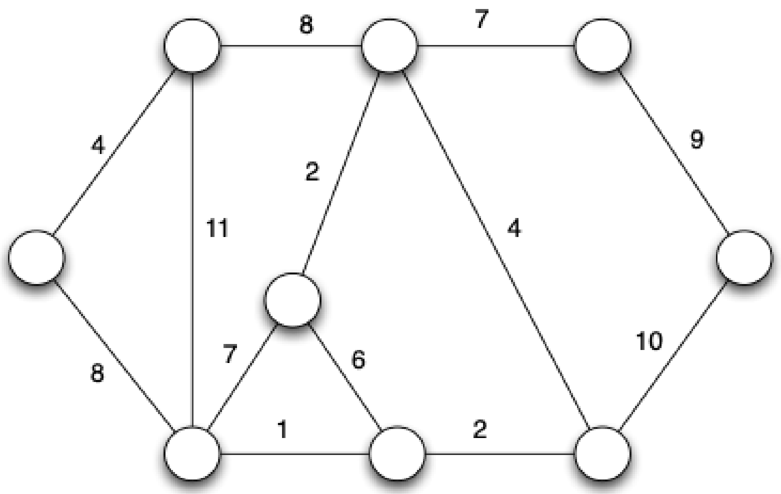
Graphe pondéré

- Un graphe pondéré G(V,E,ω) est un graphe où un entier positif est affecté à chaque arête.
- On appelle cet entier poids de l'arête.



Poids d'un graphe

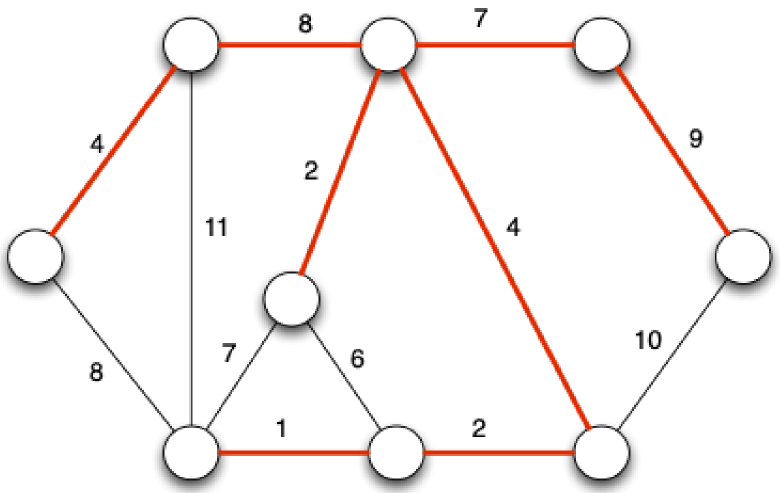
- Le poids (où coût) d'un graphe est la somme des poids des arêtes du graphe.
- On le note ω(G)



$$w(G) = 103$$

Arbre couvrant de poids minimum

- Soit un graphe G=(V,E,ω) un graphe non orienté pondéré.
- On appelle arbre couvrant de poids minimum (ou maximum) de G
    - noté ACPM ou MST (minimum Spanning Tree)
    - Tout arbre couvrant dont la somme des poids des arêtes le constituant est minimal (maximal).



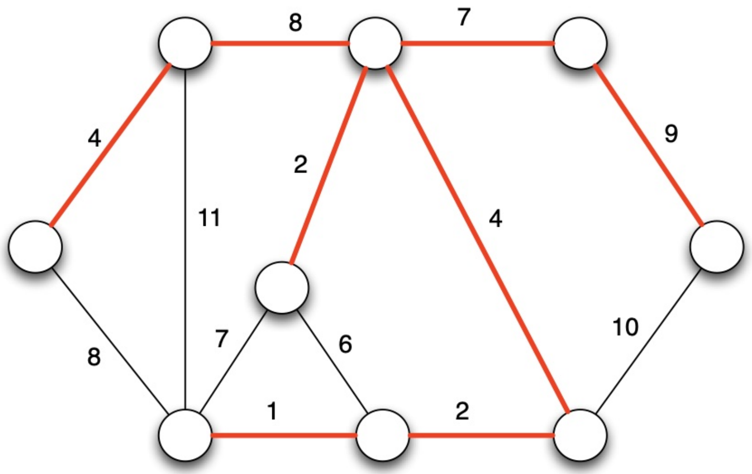
$$w(G') = 37$$

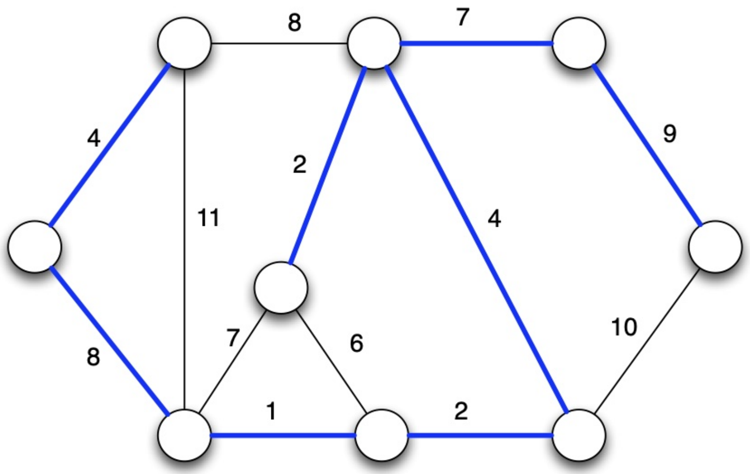
Proposition

- Un graphe admet un arbre couvrant si et seulement si il est connexe.

Remarque1

- L'arbre couvrant de poids minimal n'est pas forcément unique.



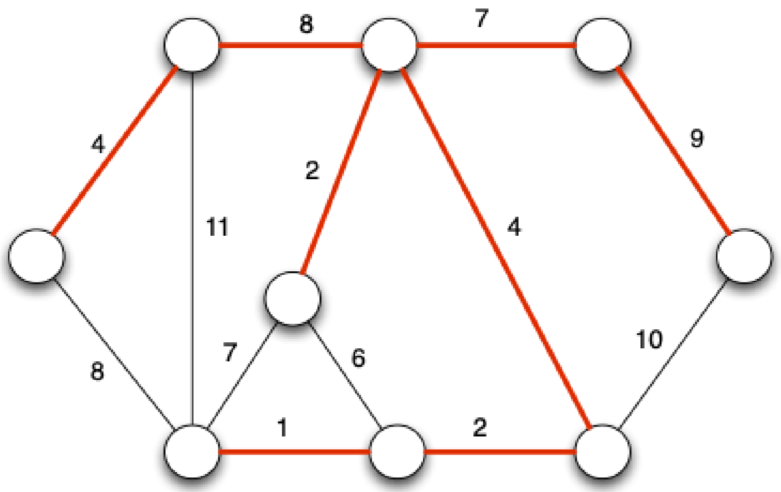


Remarque 2

- Un arbre couvrant de poids minimum est unique si et seulement si les poids de ces arêtes sont deux à deux distincts.

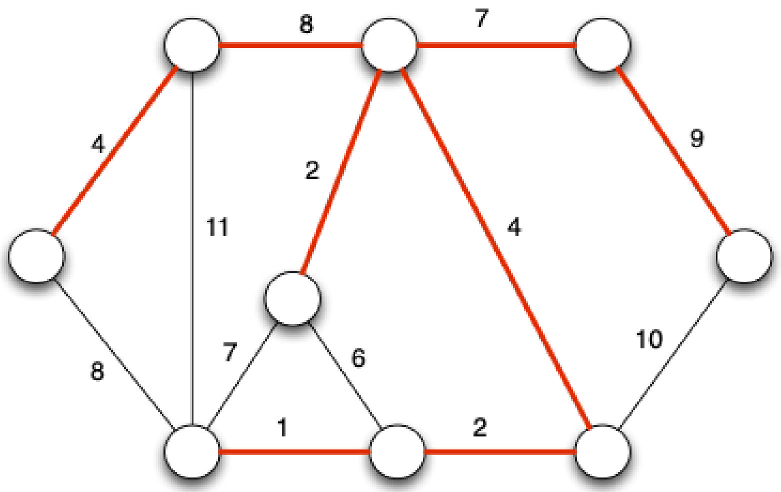
Propriété Cycle-Max

- L'arête de poids maximum d'un cycle ne fait partie d'aucun arbre couvrant de poids minimum.



Propriété Coupe-Min

- L’arête de poids minimum d’une coupe fait partie de l’arbre couvrant de poids minimum.



Algorithmes séquentiels

- Algorithme de Borůkva (1926) (Coupe-Min)
- Algorithme de Prim (1957) (Coupe-Min)
- Algorithme de Kruskal (1956) (Cycle-Max)
- Algorithme de Solin (1961) (Coupe-Min)
- ...

Question

- Comment construire de façon distribué un arbre couvrant de poids minimum?

Kruskal distribué?

- L'approche de Kruskal est-elle distribuable?

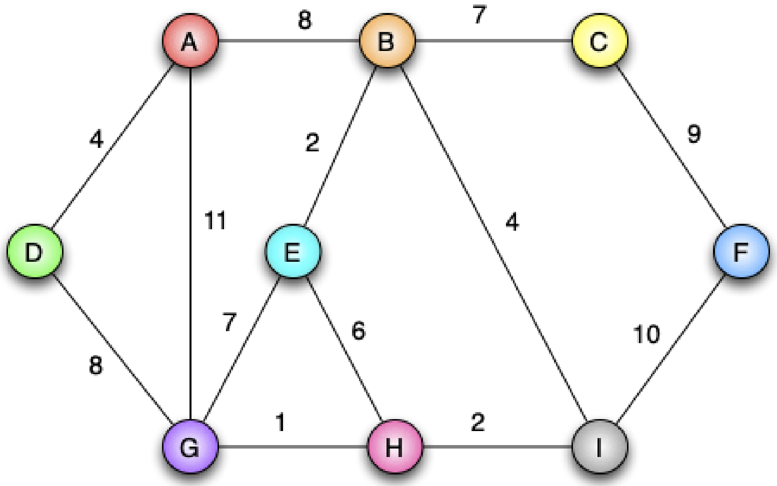
Prim distribué?

- L'approche de Prim est-elle distribuable?

Algorithme de Borůkva

- Initialement chaque sommet est une composante connexe
- Tant qu'il existe plusieurs composantes connexe:
  - Choisir une composante connexe C
  - Trouver l'arête sortante de C de poids minimum: notée e
  - Soit S la composante connexe de l'autre extrémité de e
  - Créer une nouvelle composante connexe:
    - $C \cup \{e\} \cup S$

Exemple de Borůkva



- Composantes={A}{B}{C}{D}{E}{F}{G}{H}{I}
- MST={}

Algorithme distribué

Passage au distribué

- Construire un algorithme pour le MST est « facile » en séquentiel beaucoup moins en distribué.

Gallager, Humblet et Spira (1983)

- C'est le premier algorithme distribué
- Squelette de tous les autres travaux
- Notion de fragment
  - Un fragment est l'ensemble des nœuds appartenant à un même sous-arbre de poids minimum.
- Nous allons voir dans ce cours une adaptation de l'algorithme de GHS83.

Algorithmes distribués pour le MST

- Initialement chaque nœud est un fragment
- Les nœuds d'un même fragment coopèrent
    - pour trouver l'arête de poids minimum sortante du fragment
- Les fragments sont fusionner entre eux
    - grâce à l'arête de poids minimum
- La démarche est répété jusqu'à
    - ce qu'il ne reste qu'un seul fragment
    - Le MST

      → Algorithme de Borůkva

Difficultés

- Quelle sont les difficultés d'une telle approche?

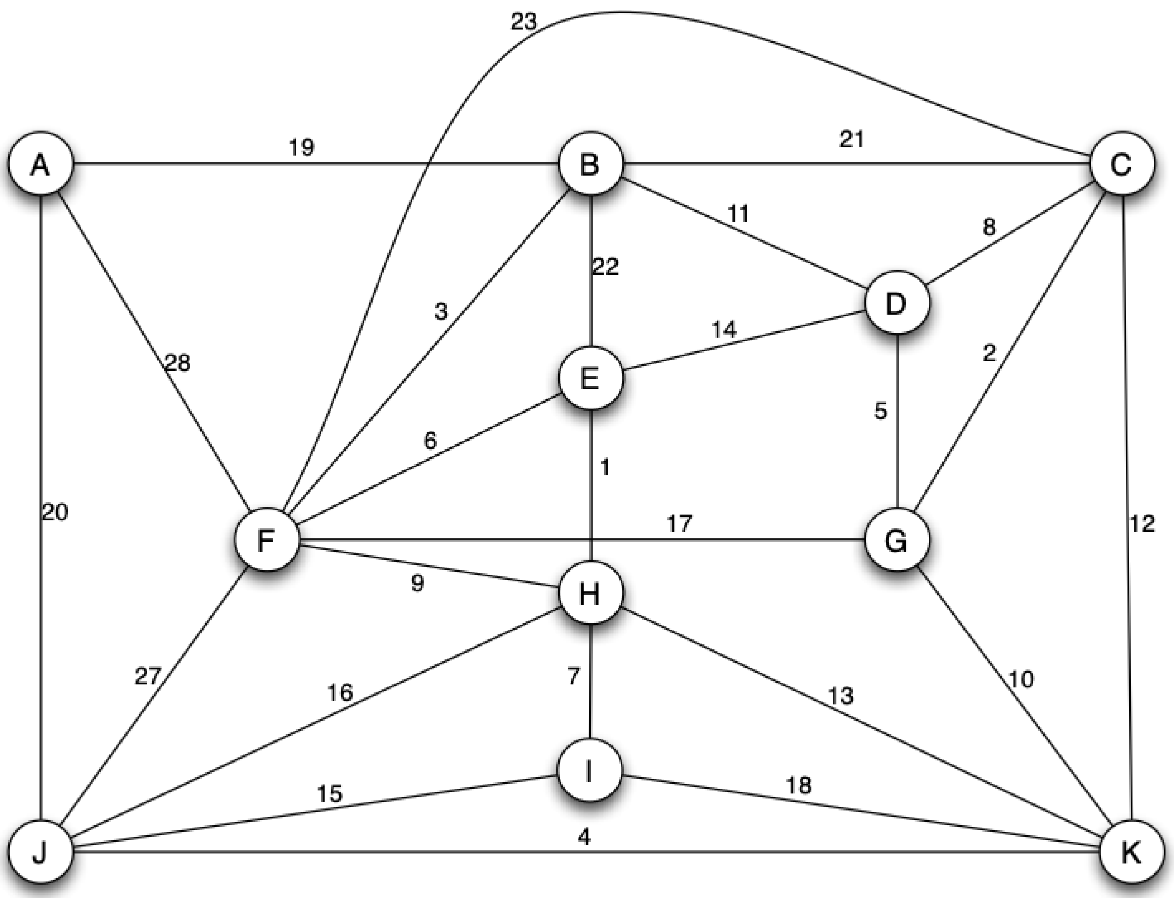
Version synchrone

1. Initialement chaque nœud est un fragment
2. Les nœuds d’un même fragment coopèrent
   - pour trouver l’arête de poids minimum sortante du fragment
3. Les fragments sont fusionner entre eux
   - grâce à l’arête de poids minimum
4. La démarche est répété jusqu’à ce qu’il ne reste
   - qu’un seul fragment
   - Le MST

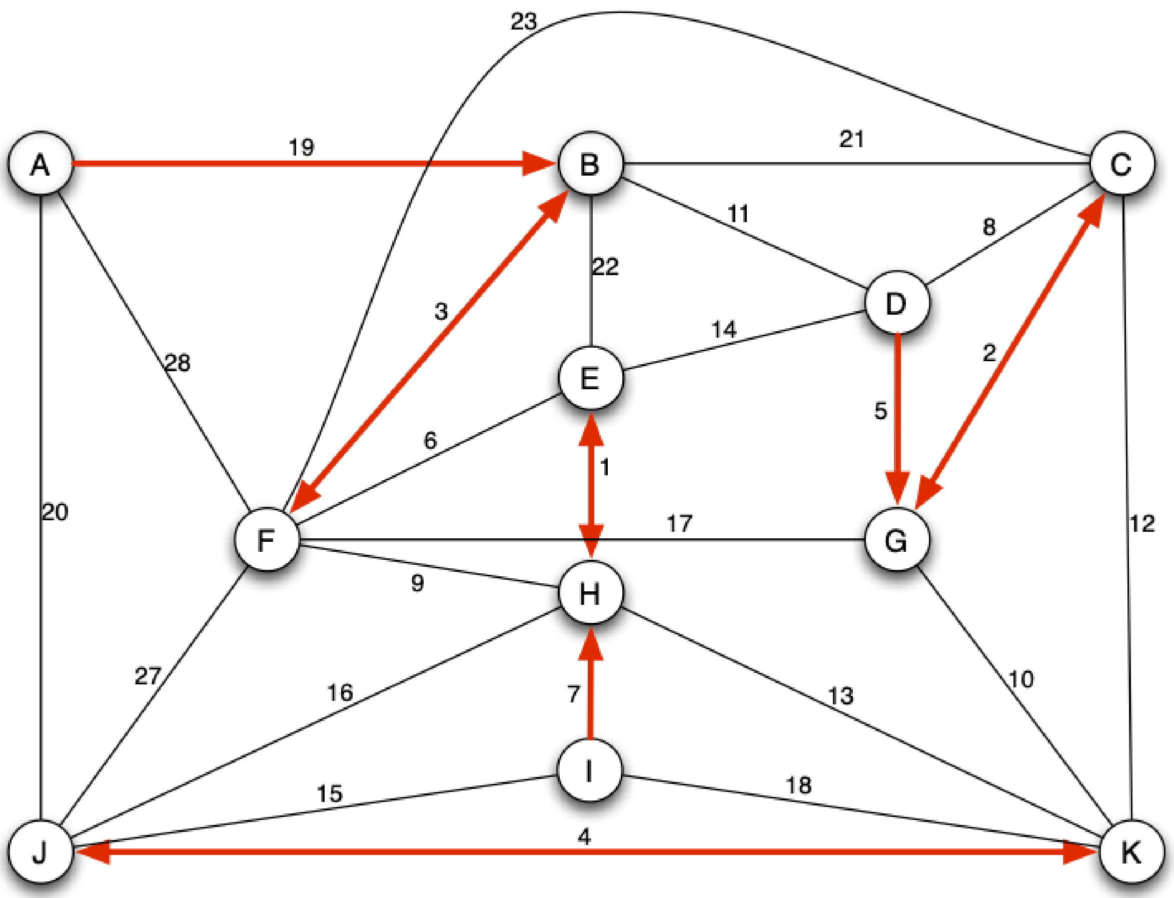
Version synchrone

- La première étape est «facile»
- Les fragments sont réduits à un seul noeud
- Donc chaque fragment peut décider localement de son arête sortante de poids minimum.

Exemple première étape



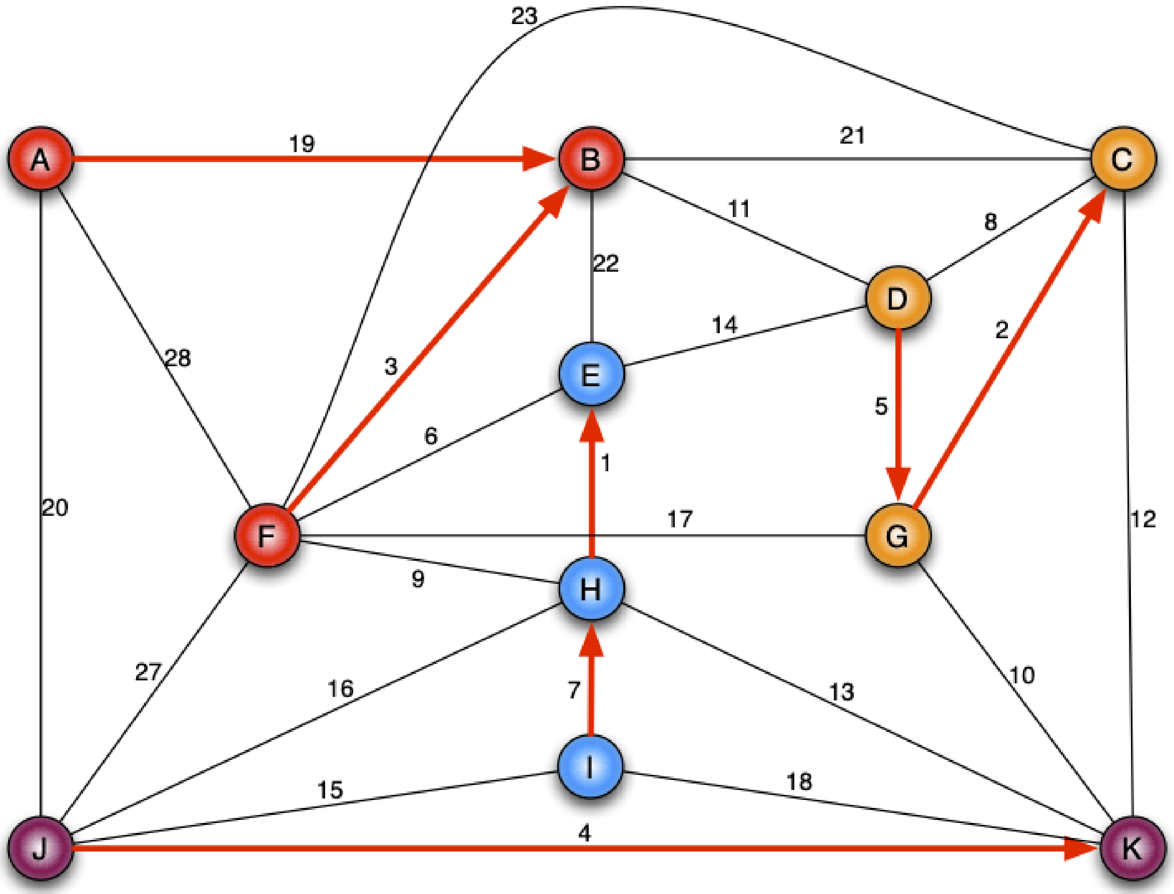
Et après la première étape



Fragments

- Un fragment est un sous-arbre couvrant orienté.
- Le parent d'un noeud est l'autre extrémité de son arête de poids minimum
- Si deux noeuds ont la même arête de poids minimum (**arête coeur**)
  - le parent est celui d'identifiant minimum
- **Comment connaitre localement son fragment?**

- La structure induite de la sélection des arêtes de poids minimum est un ensemble de sous-arbre.
- La racine de chaque sous-arbre diffuse son identifiant



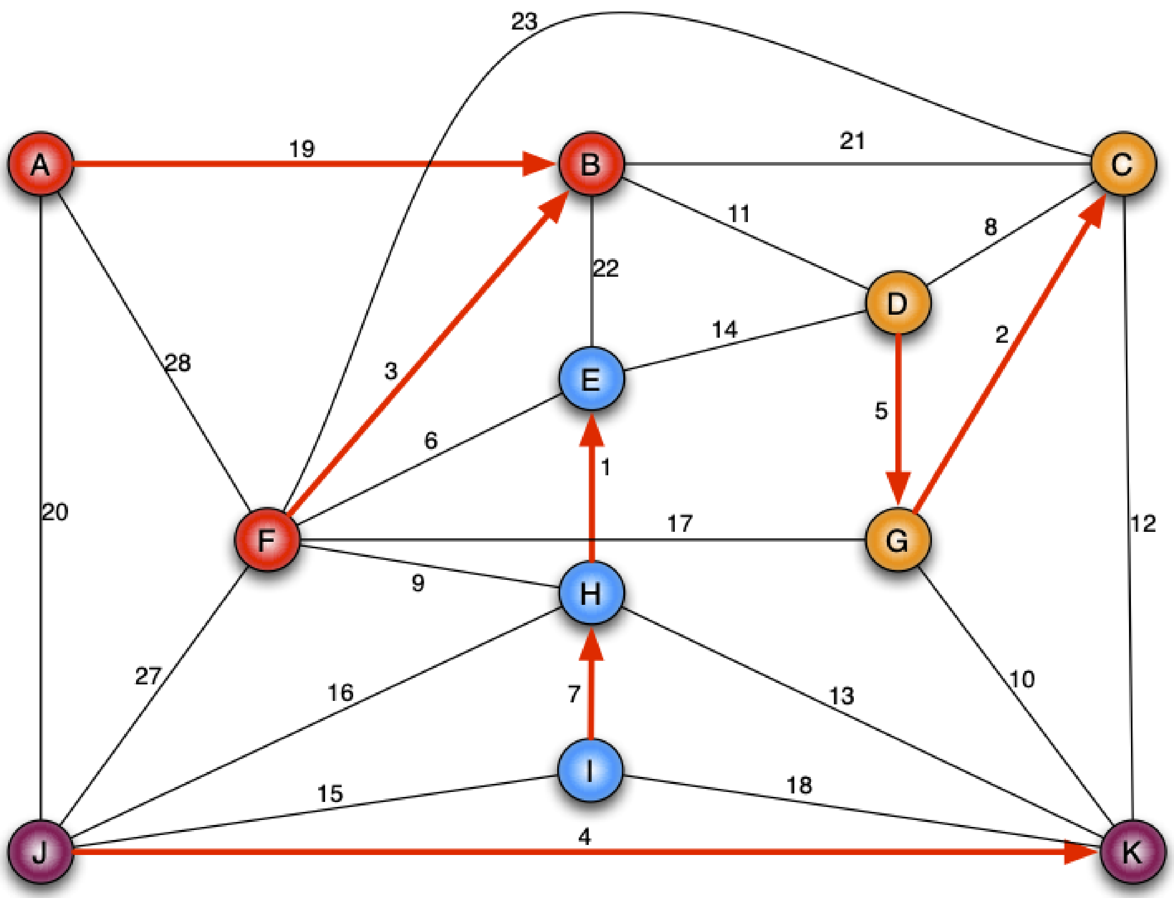
Arêtes sortantes

- Maintenant chaque noeud connait l'identifiant de son fragment
- Chaque noeud peut donc identifier localement
  - l'ensemble des arêtes internes au fragment
  - l'ensemble des arêtes sortantes du fragment
- **Comment choisir l'unique arête de poids minimum?**

Arête sortante de poids minimum

- Comment choisir l'unique arête de poids minimum?
- En partant des feuilles
  - Les noeuds sélectionnent l'arête de poids minimum sortante de leur sous-arbre.
  - La racine concentre donc l'arête sortante de poids minimum du fragment

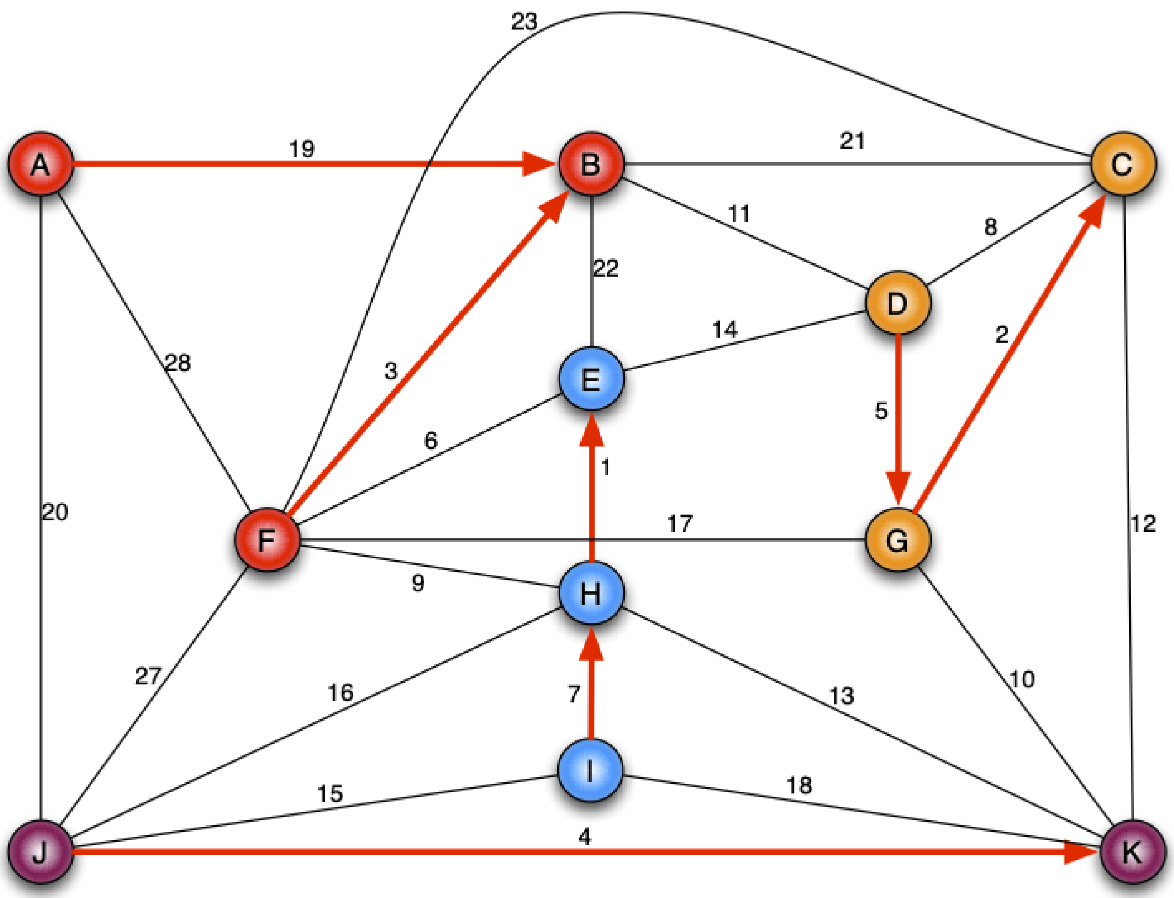
Arête sortante de poids minimum



Comment fusionner deux fragments

- Les fragments f_1 et f_2 sont fusionnés grâce à l'arête de poids minimum $e = (u, v)$
  - avec $u \in f_1$ et $v \in f_2$
- La racine du fragment f_1 est déplacer au noeud u
  - en réorientant le chemin de f_1 vers u
- Idem pour le fragment f_2

Fusion


23
A
19
B
21
C
11
8
22
D
28
3
14
2
E
5
6
1
20
17
12
F
G
9
H
27
16
13
10
7
I
15
18
J
4
K

Mise à jour des fragments

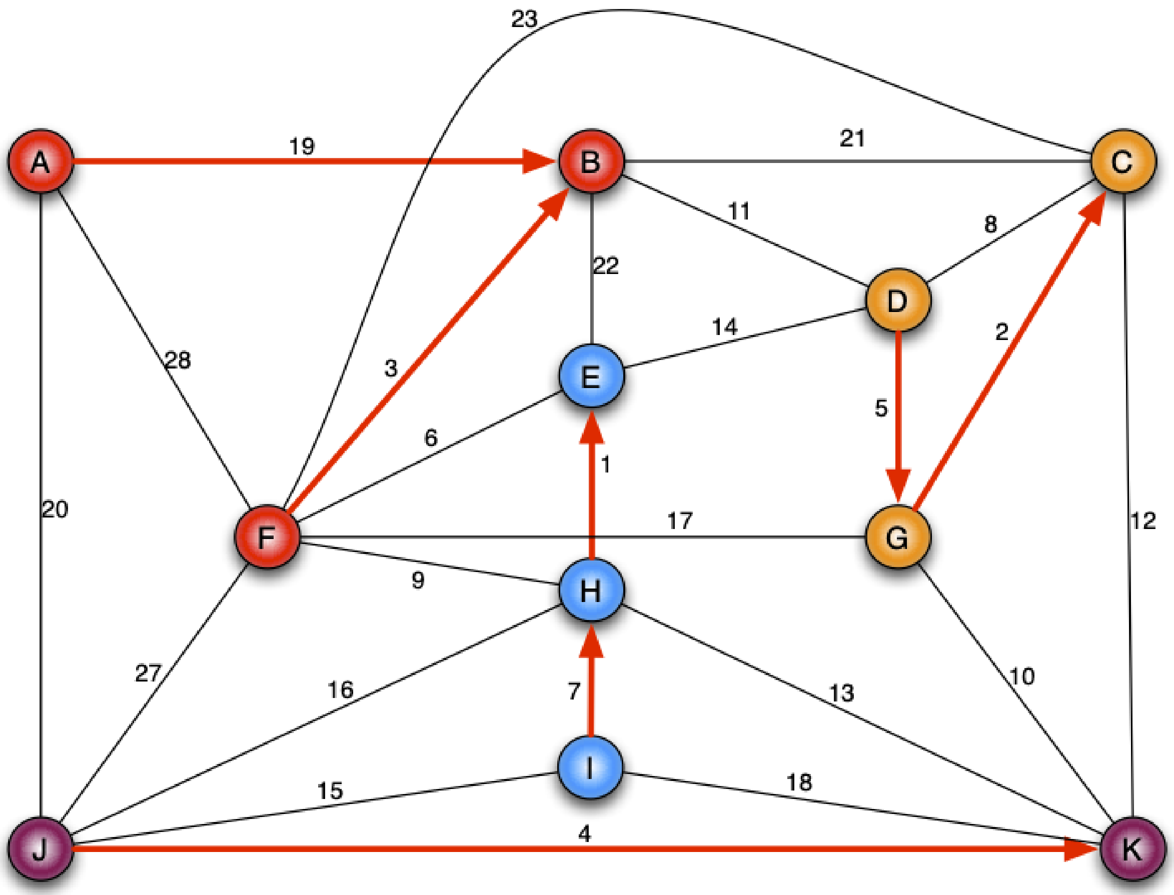
- Comment être sur que la mise à jour des fragments est terminé?

Mise à jour des fragments

- Comment être sur que la mise à jour des fragments est terminé?
- Un noeud sait qu'il à mis à jour son fragment car il a effacé son arête de poids minimum.
- Quand démarrer la collecte de l'arête de poids minimum?
- Comment être sur que les voisins ont fini leurs mise à jour?

Collecte de l'arête de poids minimum

- Fin de la fusion
  - Il efface son arête de poids minimum
  - Il envoie l'id de son nouveau fragment à ses voisins (non enfants)
  - Il efface l'es ID des fragments de ces arêtes sortantes
  - Les voisins font de même


23
A
19
B
21
C
11
8
22
D
28
3
E
14
2
6
5
1
20
F
17
G
12
9
H
27
16
7
13
10
15
I
18
J
4
K

Terminaison

- l'algorithme s'arête quand la racine ne collecte aucune arête sortante

Complexités

- $5n\log_2 n + 2m$ messages → $O(n\log_2 n + m)$ msg
  - n est le nombre de noeuds et m le nombre de liens
- Chaque message est de taille $O(log_2 W)$ bits
  - W est le poids maximum sur les arêtes

Remarques

- Cet algorithme fonctionne en asynchrone.
- Dans la version présenté les subtilités liés à l'asynchrone on été masqués.
- Il faut notamment que les fragment « grossissent » à la même « vitesse » pour minimiser le nombre de message.